On Representing Biological Systems through Multiset Rewriting

F. Martinelli, S. Bistarelli, I. Cervesato, G. Lenzini, R. Marangoni

CNR
Iit
Consiglio Nazionale delle Ricerche - Pisa
Istituto di Informatica e Telematica

Outline

- MSR for modeling bio-chemical systems
  - Stochastic MSR (MSR)
- MSR as low level language for:
  - Petri nets
  - Pi-calculus
- Concluding remarks
  - Future work

MSR

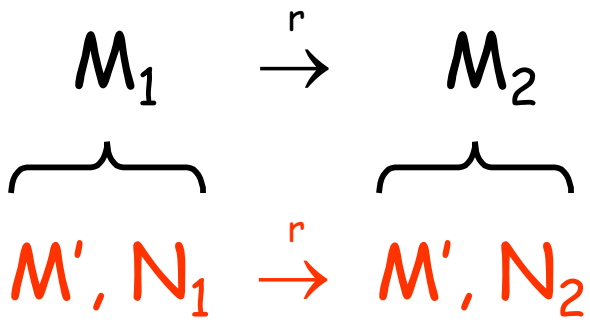
- Executable specification language
  - Similar to biochemical equations
  - Solid logical foundation
- Ties with linear logic
- Flexible and fully precise

Multiset rewriting …

- Multiset: set with repetitions allowed
- Rewrite rule:

$$r: N_1 \rightarrow N_2$$

- Application



- Multi-step transition, reachability

… with terms

- msets of 1st-order atomic formulas
- Rules:

$$r: F(\underline{\mathbf{x}}) \rightarrow G(\underline{\mathbf{x}})$$

- Application

$$\underbrace{M_1}_{M', F(\underline{t})} \xrightarrow{r} \underbrace{M_2}_{M', G(\underline{t})}$$

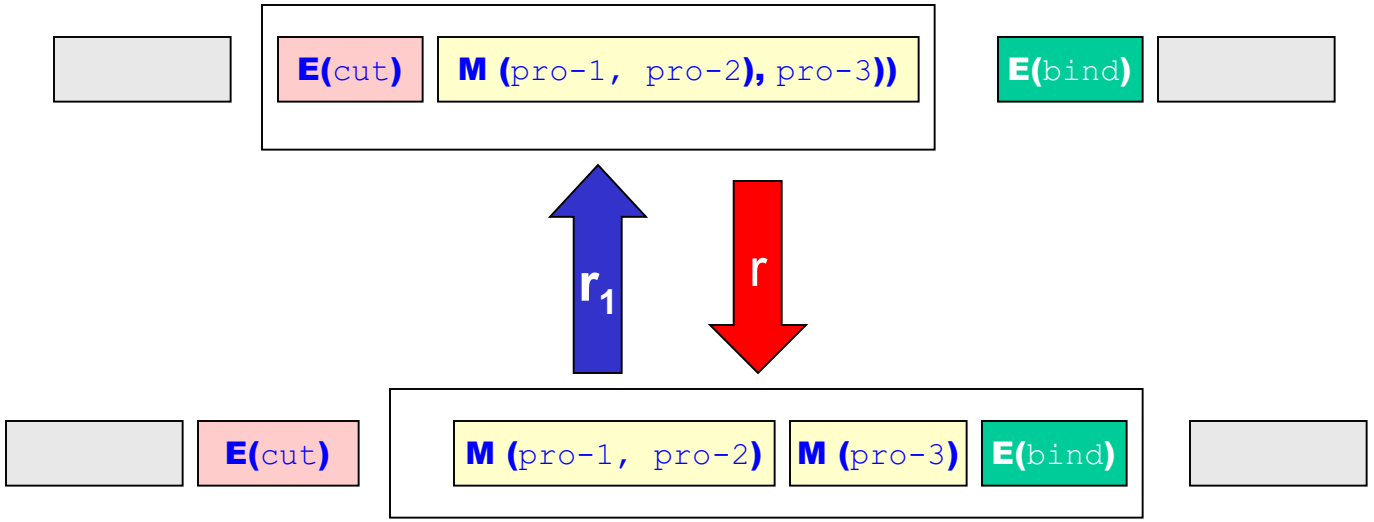
MSR-Bio simple analogy

- Predicate names → molecular species
- Predicate → molecule
- Rewrite step → reaction

MSR Examples (molecular reactions)

r : **E**(cut), **M**(p, c) → **E**(cut), **M**(p), **M**(c)

r_1 : **E**(bind), **M**(p), **M**(c) → **E**(bind), **M**(p, c)

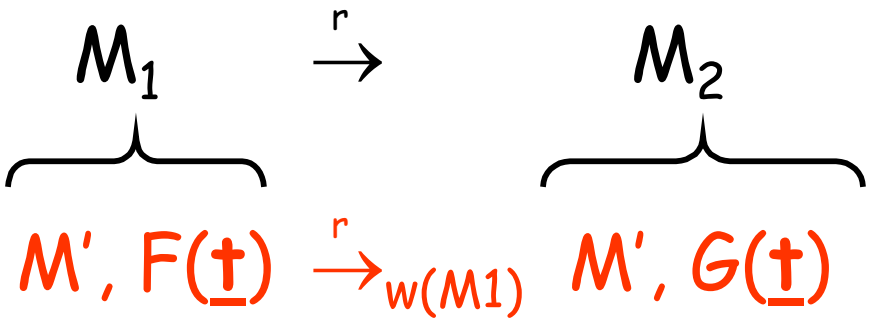


… with weights

- A weight function *w()* that depends on the current mset
- Rules:

$$r: F(\underline{\mathbf{x}}) \rightarrow_w G(\underline{\mathbf{x}})$$

- Application



MSR-Bio simple analogy (2)

- Predicate names → molecular species
- Predicate → molecule
- Rewrite step → reaction
- Weight → reaction rate

An example: dimerization of a molecule R

- Consider the biochemical equation:
  - $2R \underset{k-1}{\overset{k}{\leftrightarrow}} R_2$
- In MSR we get the two rules:
  - $R, R \rightarrow_{Wk} R_2$
  - $R_2 \rightarrow_{Wk-1} R, R$

where w_k, w_{k-1} are two weight functions that depend on the kinetic constants k, k^{-1}

MSR as a low level language

- MSR good for describing static and dynamic features of bio-chemical reactions
  - Eker et al, PSB'02.
  - Lange et al., 1999.
- Languages from concurrency theory have been adopted
  - Petri Nets [Goss-Peccoud, 98]
  - Process algebras [Priami et al., 2001]
  - Ambient calculus [cardelli et al.]

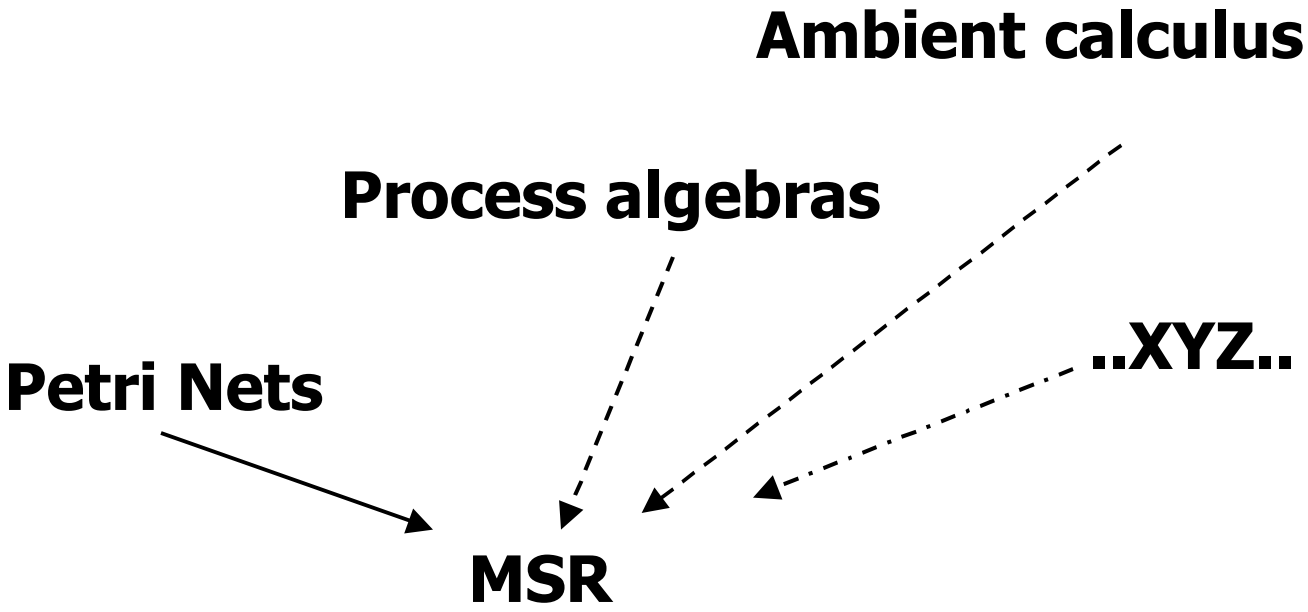
Experience from security protocols analysis (1)

- Many different description languages for security protocols
  - MSR, ambient calculus, Petri nets, process algebras, …
- Each of them very expressive in a specific framework
  - Useful for specification!
- However, this demands for a lot of research on the different analysis frameworks

Experience from security protocols analysis (2)

- A successful approach [Millen, 98]:
  - Map all such a languages on a unique simple intermediate one, i.e., MSR
  - Focus the efforts on such a simple language
  - Powerful analysis algorithms for the simple MSR rewriting rule
  - Comparison of relative merits among different description frameworks

.. Also for biological systems..



Stochastic Petri Nets as Stochastic MSR

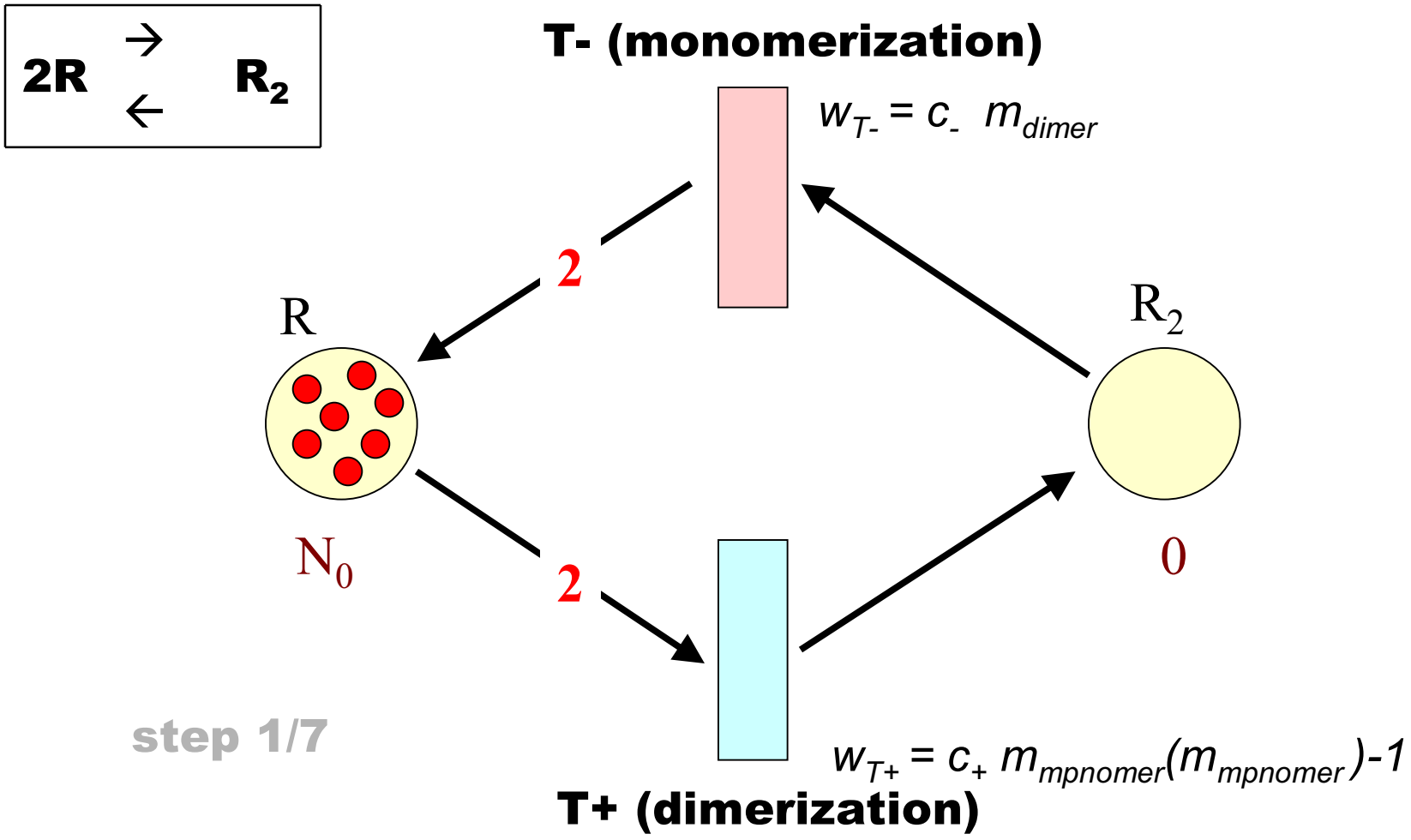
- A Petri net consists of:
  - *Places* (denote different types of resources)
  - *Tokens* (denote resources in places)
  - *Transitions* (denotes how tokens are consumed/produced). Stochastic Petri Nets have weights on the transitions
  - *Markings* (denote the current resource allocation)

Analogy SMSR - SPN

- Predicate names → Places
- Predicate → token
- Rewriting rule → transition
- Mset → marking

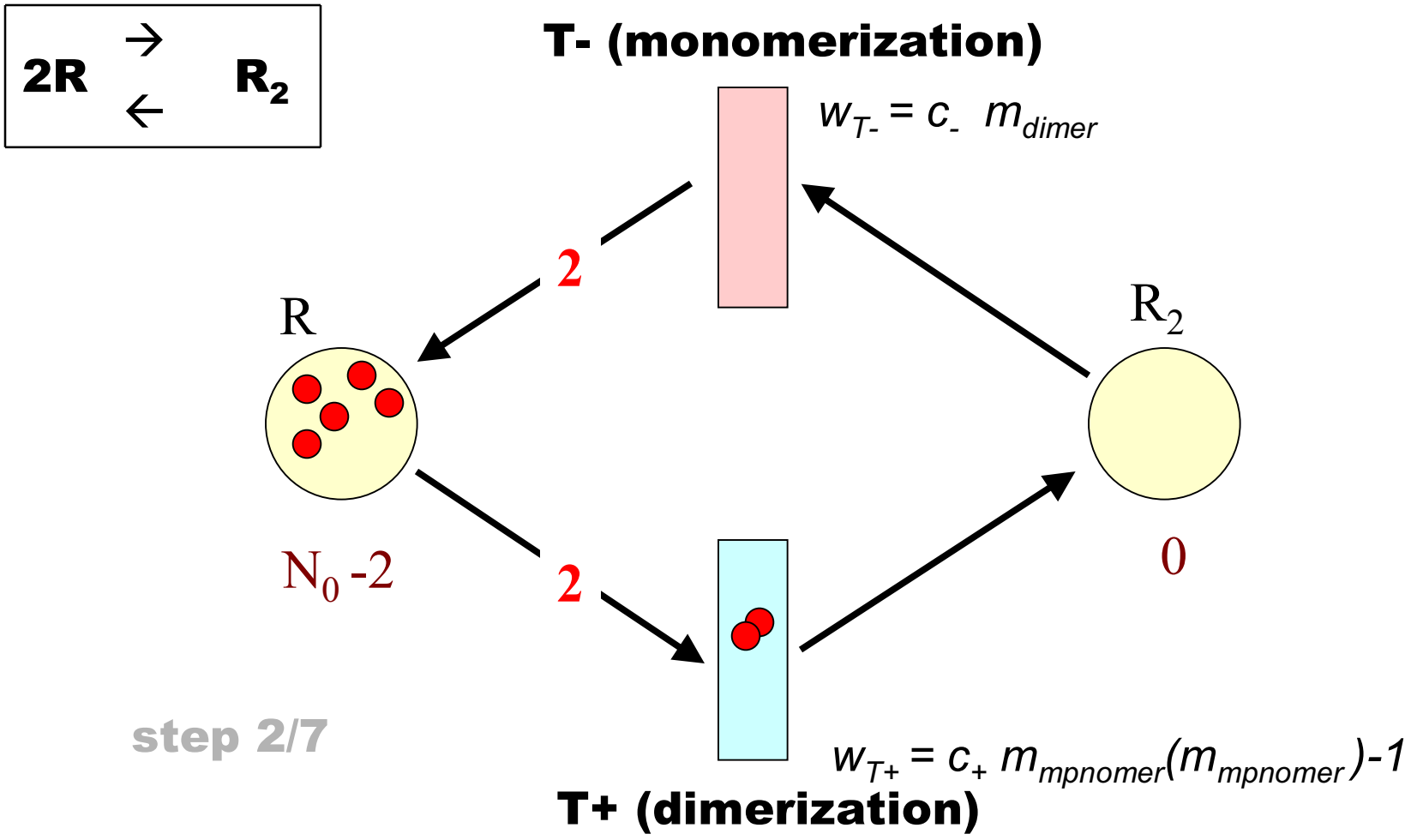
SMSR specification are closer to bio-chemical equation than SPN ones!

Dimerization example with SPN

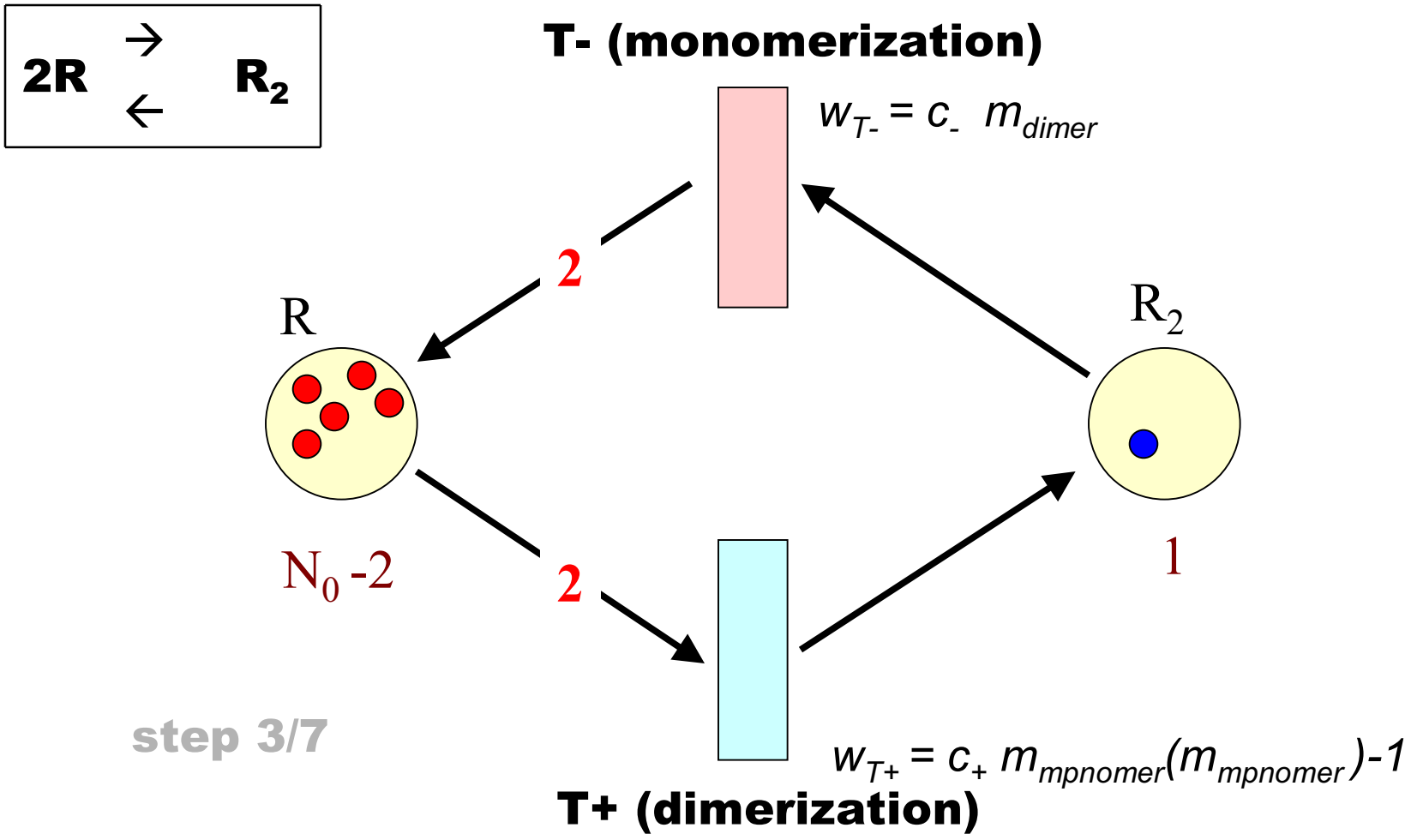


step 1/7

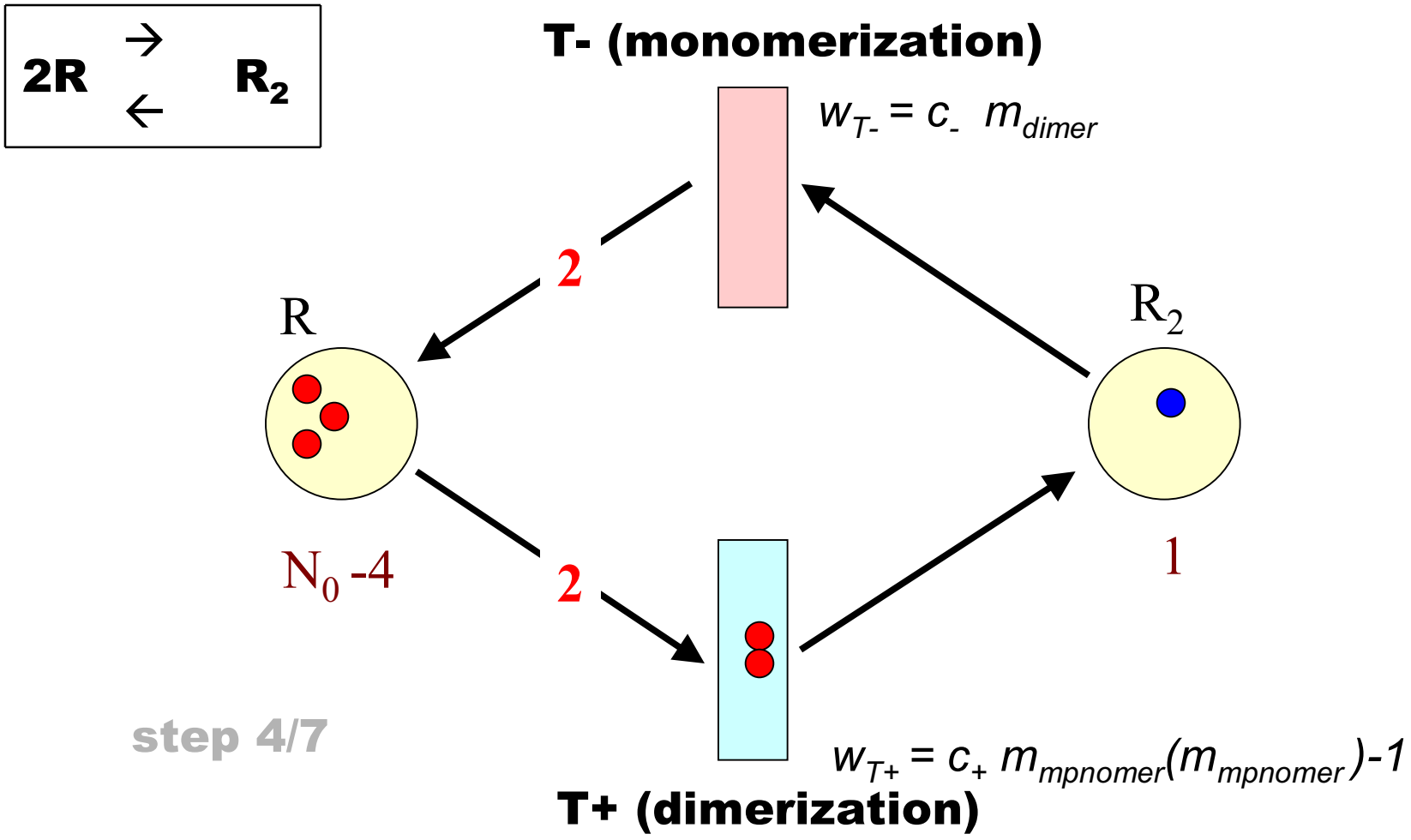
Dimerization example with SPN



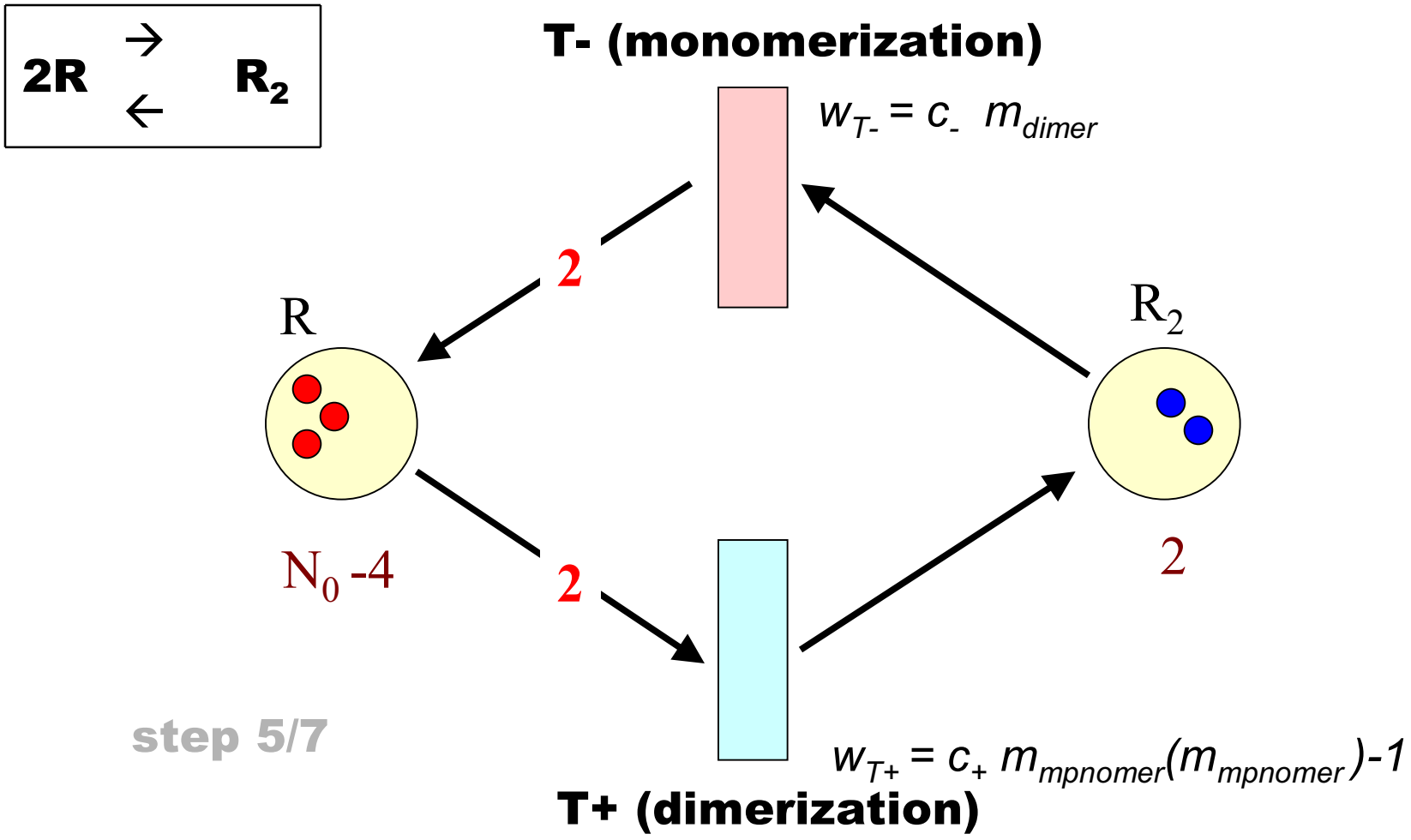
Dimerization example with SPN



Dimerization example with SPN

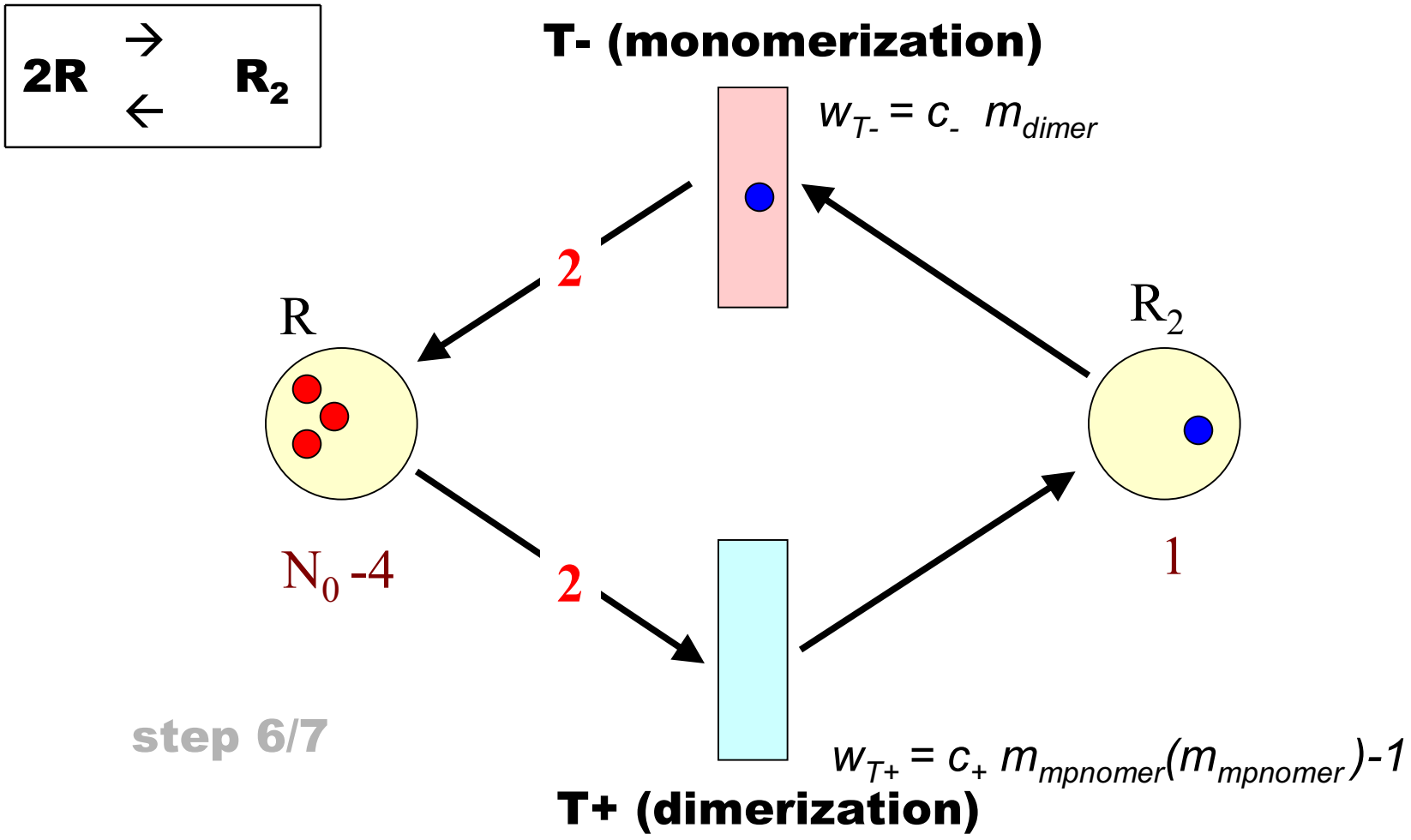


Dimerization example with SPN

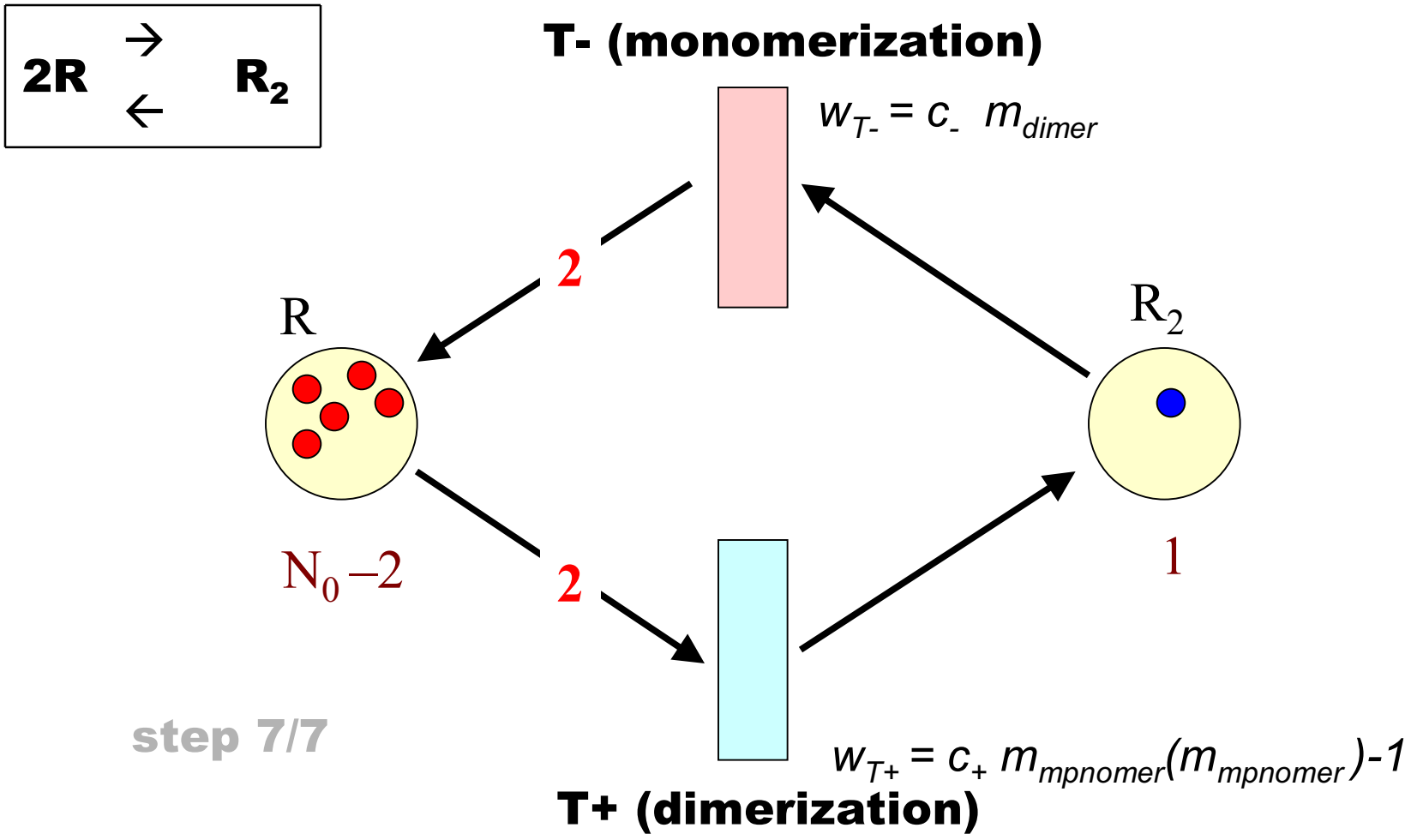


step 5/7

Dimerization example with SPN



Dimerization example with SPN



MSR vs. petri Nets

- From a specification point of view MSR seem preferable to Petri Nets due to their resemblance with biochemical equations

Process algebra as MSR

- A process algebra consists of several operators:
  - Nil operator: 0
  - Prefix, i.e. output(a,P), Input(a,P)
  - Parallel composition operator: P|P
  - Replication operator !P

Reduction semantics

- Output(a,P) $\Rightarrow$ a | P
- a | input(a,P) $\Rightarrow$ P
- !P $\Rightarrow$ P | !P
- We consider a structural equivalence among terms $\equiv$ with the rule
  - P $\equiv$ P′ and P′ $\Rightarrow$ P″ then P$\Rightarrow$ P″

Encodings of PA into MSR

- Each sequential process is mapped into a set of rules
  - [P=Input(a,P')] = ([P']$\cup r_P$)
    - Where r_P = O(a), Pre(P) -> Pre(P')
  - [P=Output(a,P')] = ([P']$\cup r_P$)
    - Where r_P = Pre(P) ->O(a), Pre(P')
  - [!P] = (. -> Pre(P)) $\cup$ [P]
  - .....

A simple example

- P=P′|P″
  - where P′= Input(a,0),P″=Output(a,0)
  - Note that P reduces to 0|0
- [P]= {r_1:Pre(P′),O(a) $\rightarrow$ Pre(0), r_2:Pre(P″) $\rightarrow$ O(a), Pre(0)}
- Applied starting from the multisiet Pre(P′),Pre(P″):
  - Pre(P′), Pre(P″) $\rightarrow r_2$
  - Pre(P′), O(a),Pre(0)$\rightarrow$r1
  - Pre(0), Pre(0)

Conclusion and ..

- We advocate the use of MSR as a low level language
  - In analogy with a current trend in security
- We performed some preliminary steps
  - SPN $\rightarrow$ SMSR
  - Process algebras (pi-calculus) $\rightarrow$ MSR [Bistarelli et al., WITS 2003]
- We argue that also ambient can be encoded into MSR:
  - Not impossible: ambient calculus $\rightarrow$ pi-calculus is a recent result of Degano-Priami

.. future work

- encoding other languages into SMSR
- Developing an efficient SMSR simulation engine
  - As done for security protocols we plan to build translators to SMSR from other languages.
    - Hopefully, we can reuse part of the ideas/code done in security
- Developing specific formal notions for SMSR applied to biology, as symbolic evaluation, abstraction, composition, equivalence ….

MSR Examples (chemical reactions)

r : **H, H, O → H2O**

r_1 : **H2O → H, H, O**